

Hindawi Publishing Corporation  
EURASIP Journal on Advances in Signal Processing  
Volume 2007, Article ID 45493, 13 pages  
doi:10.1155/2007/45493

## Research Article

# Distributed Source Coding Techniques for Lossless Compression of Hyperspectral Images

Enrico Magli,<sup>1</sup> Mauro Barni,<sup>2</sup> Andrea Abrardo,<sup>2</sup> and Marco Grangetto<sup>1</sup>

<sup>1</sup> Center for Multimedia Radio Communications (CERCOM), Department of Electronics, Politecnico di Torino, 24 Corso Duca degli Abruzzi, 10129 Torino, Italy

<sup>2</sup> Dipartimento di Ingegneria dell'Informazione, Università di Siena, Via Roma 56, 59100 Siena, Italy

Received 10 February 2006; Revised 18 October 2006; Accepted 23 October 2006

Recommended by Yap-Peng Tan

This paper deals with the application of distributed source coding (DSC) theory to remote sensing image compression. Although DSC exhibits a significant potential in many application fields, up till now the results obtained on real signals fall short of the theoretical bounds, and often impose additional system-level constraints. The objective of this paper is to assess the potential of DSC for lossless image compression carried out onboard a remote platform. We first provide a brief overview of DSC of correlated information sources. We then focus on onboard lossless image compression, and apply DSC techniques in order to reduce the complexity of the onboard encoder, at the expense of the decoder's, by exploiting the correlation of different bands of a hyperspectral dataset. Specifically, we propose two different compression schemes, one based on powerful binary error-correcting codes employed as source codes, and one based on simpler multilevel coset codes. The performance of both schemes is evaluated on a few AVIRIS scenes, and is compared with other state-of-the-art 2D and 3D coders. Both schemes turn out to achieve competitive compression performance, and one of them also has reduced complexity. Based on these results, we highlight the main issues that are still to be solved to further improve the performance of DSC-based remote sensing systems.

Copyright © 2007 Enrico Magli et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1. INTRODUCTION

In recent years, distributed source coding (DSC) has received an increasing attention from the signal processing community as a new paradigm to code statistically dependent sources [1, 2]. DSC considers a situation in which two or more statistically dependent data sources must be encoded by two separate encoders that are not allowed to talk to each other, that is, each encoder sees only the output of one of the two sources; in the following we will use the terms “dependent” and “correlated” interchangeably. Data sources must be encoded by two separate encoders that are not allowed to talk to each other, that is, each encoder *sees* only the output of one of the two sources. Following the standard encoding paradigm, each source can be compressed losslessly, with a total rate no less than the sum of the two source entropies. This is clearly less efficient than an encoder that jointly compresses the two sources, since in this latter case a bit rate equal to the joint entropy of the sources could be used. The surprising result of DSC theory [3–5] is that, under certain assumptions, the same result can be achieved by using two separate

encoders, provided that the two sources are decoded by a joint decoder. For example, it is possible to perform *standard* encoding of the first source (called side information) at its entropy, and *conditional* encoding of the second one at a rate lower than its entropy; no information about the first source needs to be available at the second encoder, but only correlation parameters such as the conditional entropy. Interestingly, DSC coders are typically implemented using channel codes; conditional encoding is performed by representing the source using its syndrome (or the parity bits) of a suitable channel code. An overview of this process is given in Section 2.1.

DSC theory can be immediately applied to all the cases where two or more correlated sources must be coded efficiently by separate encoders and decoded by a unique decoder, as is the case with distributed sensor networks applications [1]. Less evident, but equally appealing, is the possibility of applying DSC principles to situations where a single source is artificially subdivided into correlated subsources that are encoded separately. In this case, the advantage is that decorrelation among the subsources is no longer required

thus considerably simplifying the encoder (though at the expenses of the decoder). Moreover, no communication between the subsources is required, at least in the ideal case, thus greatly reducing the communication bandwidth of the processing system. These concepts have a strong potential for remote sensing image compression; this application field has been devoted preliminary investigations in [6–9]. In a remote sensing system, correlation among different images can be found at several different stages. Some potential applications include (i) exploiting the correlation between two (or more) bands of a multispectral or hyperspectral image to achieve lower encoder complexity by avoiding explicit decorrelation, (ii) exploiting the correlation between newly acquired and archived images, (iii) exploiting the correlation between multimodal images, (iv) exploiting the correlation between archived images to reduce storage space. The first application is the one dealt with in this paper.

Although several near-capacity DSC coders have been designed for simple ideal sources (e.g., Gaussian sources), the applications of practical DSC schemes to realistic signals have been so far quite limited, and have yielded somewhat controversial results. On one hand, the performance of practical systems is often further away from the theoretical bounds than in the ideal case due to various characteristics of the signal to be coded that do not fit the ideal model. On the other hand, the complexity required to achieve these results may outweigh the computational saving obtained by reducing the decorrelator complexity; moreover, additional system-level requirements (such as the availability of a feedback channel) may be posed in order to make DSC feasible at all. These issues are discussed in more detail in Section 2.3.

The goal of this paper is to evaluate the performance of DSC coders in a realistic application environment, such as hyperspectral image compression. In particular, we design two DSC coders based on different principles, and assess the potential of either coder to accomplish the advantages of DSC, that is, moving complexity from the encoder to the decoder by following an approach similar to that proposed in [10] for the compression of video sequences. The first codec employs powerful error-correcting codes, as is typically done in most existing DSC schemes; the second codec employs scalar multilevel coset codes. Some parts of both coders borrow from existing schemes, while their integration in a compression algorithm for hyperspectral images presents a few innovative aspects. The new codecs have been tested on a set of scenes of the AVIRIS sensor, and the results have been compared with those achieved by some popular 2D and 3D compression algorithms (JPEG-LS, 2D and 3D CALIC). The results we obtained provide useful insights about the challenges to be solved to effectively apply DSC principles to practical scenarios.

The remainder of the paper is organized as follows. In Section 2, the basic results of DSC theory are presented and prior works attempting to turn these results into practical algorithms are reviewed. In Section 3, the first DSC codec we developed is described and its performance is discussed in Section 4. A second algorithm, following an alternative approach with complementary advantages and drawbacks, is

presented in Section 5 and the corresponding experimental analysis is given in Section 6. The paper ends with Section 7, where some conclusions are drawn and directions for future research are indicated.

## 2. DISTRIBUTED SOURCE CODING: OVERVIEW AND STATE OF THE ART

### 2.1. Overview of syndrome-based coding

We consider two correlated information sequences  $[\dots, X_{-1}, X_0, X_1, \dots]$  and  $[\dots, Y_{-1}, Y_0, Y_1, \dots]$  obtained by repeated independent drawing from a discrete bivariate distribution  $p(x, y)$ . We denote these sequences as  $X = \{X_k\}_{k=-\infty}^{\infty}$  and  $Y = \{Y_k\}_{k=-\infty}^{\infty}$ . If two separate encoders are used, the total rate required to represent the two sources exactly is  $R_{\text{tot}} = R_x + R_y \geq H(X) + H(Y)$ , where  $H(\cdot)$  denotes entropy. If the intersource correlation was exploited, for example, by means of a joint encoder that provides a single description of both  $X$  and  $Y$ , the total rate would be lower-bounded by the joint entropy, that is,  $R_{\text{tot}} \geq H(X, Y)$ . In DSC, two separate encoders generate descriptions of  $X$  and  $Y$ , and a *joint* decoder reconstructs the pair of signals. Slepian and Wolf [3] have shown that, somewhat surprisingly, such scheme can theoretically achieve the same asymptotic performance as the joint scheme.

In most existing DSC schemes, this is achieved by means of “binning.” To explain this concept, let us assume that  $X$  and  $Y$  are binary strings, and that the block size is not infinity but a given  $n$ . The set of  $2^n$  possible values assumed by  $X$  can be partitioned into  $2^{n-k}$  cosets  $\mathcal{C}_i$ ,  $i = 1, \dots, 2^{n-k}$ , containing  $2^k$  elements each;  $k$  is chosen so that  $(n - k)/n \gtrsim H(X | Y)$ . When the encoder receives a signal  $X$ , it seeks the coset  $\mathcal{C}^*$  to which  $X$  belongs. Instead of coding  $X$  at rate  $H(X)$ , the encoder sends the label that identifies  $\mathcal{C}^*$ ; this requires  $n - k$  bits. The joint decoder compares the elements in  $\mathcal{C}^*$  with the side information  $Y$ , and picks as estimate  $X^*$  the element of  $\mathcal{C}^*$  that is closest to  $Y$ . Note that designing such coset code requires prior knowledge of  $H(X | Y)$ ; in practice this may be an issue.

Typically, the input alphabet is partitioned by means of a linear channel code, in such a way that all messages with the same syndrome (here playing the role of coset label) are assigned to the same coset. Syndrome-based coding of a binary source  $X$  involves the use of an  $(n, k)$  linear channel code with  $(n - k) \times n$  parity-check matrix  $\mathcal{H}$ . Using this channel code for error correction, the length- $k$  message  $X$  is transformed into a length- $n$  message  $X'$  by appending  $n - k$  parity bits to  $X$ . Using it as a source code, the message  $X$  has length  $n$ , and  $n - k$  syndrome bits are computed as  $S = \mathcal{H}X$ . The channel code rate is defined as  $k/n$ , whereas the obtained compression ratio is  $n/(n - k)$ .

### 2.2. Prior work

Several practical techniques have been proposed for Slepian-Wolf (S-W) and Wyner-Ziv coding of Gaussian sources. If the sources are not i.i.d. in the sense of Section 2.1, a decorrelator

must be applied. The distributed Karhunen-Loeve transform [11] is the optimal decorrelator; however, in practice standard transforms and predictors have been used in all existing works, due to the difficulty of modeling the inter-source correlation as required by the optimal transform. As for the coding stage, it has been shown (see, e.g., [12]) that channel codes that are able to get very close to channel capacity are good candidates for S-W coding. The intuition is that, if the correlation between  $X$  and  $Y$  can be modeled as a “virtual” channel described as  $X = Y + W$ , a good channel code for that transmission problem is also expected to be a good S-W source code. Therefore, S-W coding is generally carried out by employing the cosets of good linear block or convolutional codes. In the lossy case, a quantizer is also employed to set the distortion, followed by an S-W entropy coder. The first practical technique has been described in [13], and employs trellis-coded quantization and trellis channel codes. Recently, more powerful channel codes such as turbo codes have been proposed in [2, 14, 15], and low-density parity-check (LDPC) codes have been used in [16]. Note that the constituent codes of turbo codes are convolutional codes, hence the syndrome is difficult to compute. In [2], the cosets are formed by all messages that produce the *same parity bits*, even though this approach is somewhat sub-optimal [15], as these cosets do not have as good geometrical properties as those of syndrome-based coding. Turbo codes and LDPC codes can get extremely close to channel capacity, although they require the block size  $n$  to be rather large. Multilevel codes, that is, codes working on alphabets of size larger than two, have also been proposed [17] in order to exploit the correlation between the multiple binary sources, for example, bit-planes, that can be obtained from a multilevel source.

In addition to coding of Gaussian signals, a few applications to real-world data have also been proposed. In [10], the trellis-based construction of [13] is applied to the video coding problem. The idea is to consider every video frame as a different source; DSC allows to encode the video *without* performing motion estimation at the encoder, as in motion JPEG, with performance similar to a video coder that performs interframe decorrelation. Hence, this scheme reverses the classical video coding paradigm, by requiring a light encoder and a complex (joint) decoder. Similar ideas have been proposed in [2] using turbo codes. In [18], a Wyner-Ziv image coding technique working in the pixel domain is proposed, while in [19, 20], a scheme preventing the performance loss in scalable video coding is proposed, which performs DSC between the base and enhancement layer of a video sequence. In [21], coset codes are used to improve error resilience for video transmission. In [6, 7], it has been proposed to apply S-W coding to hyperspectral data in order to obtain a light on-board encoder; in [6] the S-W coder is based on LDPC codes, whereas in [7] it employs a scalar code. In [8], Wyner-Ziv wavelet-based coding of hyperspectral data is investigated, and improvements with respect to SPIHT [22] are shown for a few AVIRIS bands; a wavelet-based technique, applied to video coding, is also proposed in [23]. A technique for Wyner-Ziv coding of multispectral

images based on set theory is proposed in [9]; the performance of this technique is worse than that of JPEG 2000 [24]. The techniques in [8, 9] deal with lossy image coding, while the subject of the present paper is lossless coding. Wavelet-based techniques can be easily extended to the lossless case using integer transforms and transmitting all bit-planes, though, for lossless compression, this approach is known to yield a performance loss with respect to prediction-based techniques [25].

### 2.3. Technical challenges

Although DSC coders based on channel coders have been shown to achieve near-optimal performance on ideal sources, the performance loss turns out to be significant on real-world data. There are several assumptions of DSC theory that are only partially satisfied by practical coders; two of them are outlined in the following, and will be discussed later on with the aid of experimental results.

Most channel codes used in syndrome-based coding are optimized for binary data, whereas typical signals are multi-level sources. Binary channel codes can be applied to 16-bit data, like most hyperspectral data, by means of a bit-plane approach, however, this approach does not exploit the inter-bit-plane correlation. Therefore, there is some performance loss to be expected by neglecting this correlation. As will be seen, we have found this loss to be significant.

The conditional entropy  $H(X | Y)$ , which determines the rate of the code to be used for  $X$ , is assumed to be known at the encoder. This assumption causes a few practical problems. Ignoring  $H(X | Y)$  requires some additional mechanism to ensure that the correct code rate has been selected. For example, in [2] a feedback channel is set up between the video encoder and decoder; punctured codes are used, and a cyclic redundancy code (CRC) is also sent to allow for detection of decoding errors. The decoder requests additional parity bits until it achieves error-free decoding. If the two signals  $X$  and  $Y$  are not physically separated, but are stored in the same memory, some interband communication may be necessary in order to estimate  $H(X | Y)$ ; if this communication does not outweigh the cost of explicit decorrelation, then it makes sense to apply DSC in order to decrease the encoder complexity at the expense of the decoder's.

### 2.4. Two proof-of-concept codecs

It is evident from the above discussion that the practical application of DSC principles is not straightforward, and several issues have to be taken into account. The most common DSC designs are based on capacity-achieving channel codes, such as LDPC [26, 27] and turbo codes [2, 28]. Irregular LDPC codes are known to almost reach channel capacity; when used as entropy coders in the DSC scenario, they are expected to provide a bit rate reasonably close to the conditional entropy of the source given the side information. However, LDPC codes are so powerful only when the data block size is very large, for example, in excess of ten thousand samples, or even more; this could make the encoding process

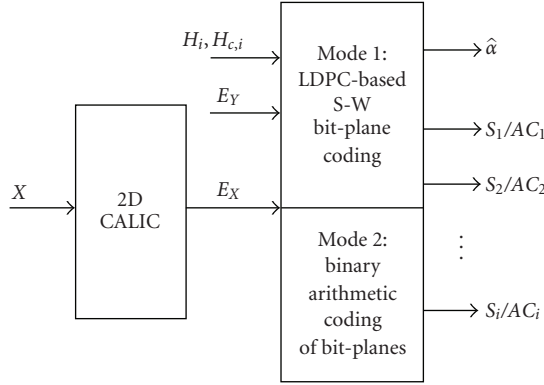


FIGURE 1: Block diagram of the proposed encoder.

computationally demanding, possibly outweighing the benefits of DSC.

Scalar codes are at the other end of the spectrum. These codes are much less powerful because they operate sample by sample; however, the encoding and decoding processes are much simpler, which is a good fit to the DSC scenario. Moreover, unlike LDPC codes, it is easy to design a scalar multi-level code, as opposed to a binary one.

In the following sections, we present and compare two different DSC algorithms, based on either approach. In particular, the first algorithm, described in Section 3, employs LDPC codes as S-W codes; its performance evaluation is reported in Section 4. The second algorithm, based on scalar codes, and amenable to a vector extension, is described in Section 5 and its results are shown in Section 6.

### 3. A DSC SCHEME FOR HYPERSPECTRAL IMAGES BASED ON CAPACITY-ACHIEVING CODES

The algorithm described in the following is called DSC-CALIC; it combines the 2D prediction stage of CALIC [29] with an S-W entropy coder to exploit interband redundancies. The S-W entropy coder is based on the syndrome-based channel coding using LDPC codes, similarly to what has been proposed in [16]; the coder is improved by the selective use of the channel code and an arithmetic coder.

As for the prediction stage, CALIC employs a nonlinear gradient-adjusted prediction that uses seven adjacent pixels in the neighborhood of the pixel to be encoded. Since this algorithm is well known, the reader is referred to [29] for further details.

Figure 1 sketches a block diagram of the proposed encoder. Let  $X$  and  $Y$  be, respectively, the current (to be coded) and previous (already coded) bands of a hyperspectral scene.

First, the CALIC encoder is applied to  $X$  in order to generate the prediction error array  $E_X$ , which is ideally a zero-mean stationary memoryless source. The prediction error of the previous band,  $E_Y$ , is available from the encoding process of the previous band, and can be used as side information at the decoder.

The next step aims at “improving” the side information for the decoder. Specifically, we employ a linear correlation model, such as  $E'_Y = \alpha E_Y$ , to obtain a modified side information  $E'_Y$  that is closer to  $E_X$ . The parameter  $\alpha$  is obtained imposing that  $E'_Y$  and  $E_X$  have the same energy:

$$E'_Y = \text{round}(\hat{\alpha} E_Y) \quad (1)$$

with

$$\hat{\alpha} = \mathcal{Q} \left[ \frac{\sum_{i=1}^N (E_{X,i})^2}{\sum_{i=1}^N (E_{Y,i})^2} \right], \quad (2)$$

where  $E_{X,i}$  and  $E_{Y,i}$ , with  $i = 1, \dots, N$ , are all the prediction error samples contained in the band being coded, and  $\text{round}(\cdot)$  denotes rounding off to the nearest integer. The operator  $\mathcal{Q}[\cdot]$  denotes scalar quantization; in fact, a 16-bit quantized version of  $\hat{\alpha}$  is employed, and is written in the compressed file to facilitate decoding.

After computing  $\hat{\alpha}$ , the actual encoding process takes place. In particular, the encoder decomposes the prediction error array  $E_X$  into its bit-planes  $E_{X,i}^{(b)}$ , with  $0 \leq i \leq \lceil \log_2(\max |E_{X,i}|) \rceil$ , plus an additional bit-plane containing the signs of the samples. Specifically,  $E_{X,i}^{(b)}$  is the  $i$ th bit-plane of  $E_X$ , and the superscript simply indicates that the data array is binary; its entropy is denoted as  $H_i = H(E_{X,i}^{(b)})$ . The equivalent bit-plane  $E_{Y,i}^{(b)}$  will be used as side information for decoding. For each bit-plane, the conditional entropy  $H_{c,i} = H(E_{X,i}^{(b)} | E_{Y,i}^{(b)})$  is assumed to be known.

For each bit-plane  $E_{X,i}^{(b)}$ , the encoder selects one out of two possible coding modes, that is, (1) encode the current bit-plane employing the DSC mode with syndrome-based coding, and (2) encode the current bit-plane using an arithmetic coder. Mode 2 is typically used when DSC is not efficient for a given bit-plane, whereas mode 1 is employed the majority of times. Mode 1 exploits the fact that DSC theory ensures that each bit-plane  $E_{X,i}^{(b)}$ , regarded as a *binary* source, can be transmitted at a rate  $R_i \geq H_{c,i}$ . In practice, mode 1 triggers the LDPC encoder and performs syndrome-based coding of the bit-plane  $E_{X,i}^{(b)}$ . The syndrome  $S_i$  of the array  $E_{X,i}^{(b)}$  is computed using an LDPC code of suitable rate, and is written in the compressed file; we select in the available database the LDPC code with highest rate, whose syndrome is no smaller than  $H_{c,i} \times N$  bits.

In particular, we use irregular LDPC codes [26, 30] with belief-propagation decoding. A database of 100 codes with code rates between 0.01 and 0.95 has been designed based on the density evolution optimization technique described in [31], for a block size of  $N = 157184$  bits (the block size is obtained as the product of the number of lines that we code as one entity, that is, 256, times the number of pixels per line). This size corresponds to a bit-plane obtained scanning in raster order all pixels of 256 lines of an AVIRIS scene. A scene containing 512 lines can be encoded in two runs; since the coding process is lossless, there is no need to worry about possible boundary artifacts caused by tiling. The performance gap of these codes with respect to ideal S-W coding is between 0.001 and 0.07 b/p.



Mode 2, which is based on the CACM [32] arithmetic coder, is used in a few specific cases.

- (i) For all the bit-planes of a few bands, for which it has been found experimentally that DSC provides little or no advantage with respect to classical 2D coding. These bands are typically the noisiest ones, for example, those corresponding to the water absorption region. Bands from 1 to 4, from 107 to 114, and from 153 to 169 are quite critical for DSC, because the difference between the entropy and the conditional entropy is very small, typically less than 0.1 b/p. Hence, mode 1 has been disabled for all bit-planes belonging to those bands (more on this can be found in Section 4). For the first band, this intraband coding mode is also dictated by the fact that no side information is available.
- (ii) For all the bit-planes for which  $H_i < 0.01$  or  $H_{c,i} > 0.95$ , because our code database does not contain any LDPC code that would outperform an arithmetic coder in those cases. The case of  $H_i < 0.01$  is typical of a few most significant bit-planes, in which most of the bits are zero, while the case  $H_{c,i} > 0.95$  is typical of the least significant bit-planes, which are very noisy and hence very difficult to compress.

Mode selection requires one bit of signaling per bit-plane per block, which represents a negligible overhead given the large block size (the worst case is about  $1 \cdot 10^{-4}$  bpp). In terms of complexity, mode selection requires estimating the entropy and the conditional entropy of each bit-plane. The conditional entropy would have to be estimated in any case in order to select the LDPC code rate; the entropy can be easily derived from the conditional entropy, hence mode selection does not generate a significant complexity overhead.

### 3.1. DSC-CALIC decoder

The decoding process works as follows. For each band, the side information  $E'_Y$  is generated by multiplying the prediction error of the last decoded band  $E_Y$  by  $\hat{\alpha}$  and rounding to the nearest integer, as in (1). Then, the side information bit-planes  $E'^{(b)}_{Y_i}$  are extracted.

The decoding process is different according to whether mode 1 or mode 2 has been used by the encoder. If mode 2 has been used, as many bits as are necessary to the arithmetic decoder to output  $N$  samples are read, and the next bit-plane is processed.

If mode 1 has been used, the decoder runs the iterative message-passing LDPC decoding algorithm, with no more than 100 iterations, to recover bit-plane  $E^{(b)}_{X_i}$ . In particular, the LDPC decoder takes as inputs the log-likelihood ratios (LLR) of  $E^{(b)}_{X_i}$ , the received syndrome of  $E^{(b)}_{X_i}$ , and the side information  $E'^{(b)}_{Y_i}$ , and performs syndrome-based decoding of  $E^{(b)}_{X_i}$ , attempting to converge to an estimated message having exactly the received syndrome  $S_i$  (see [27] for details of the LDPC decoding process).

At the heart of the decoding process are the LLRs for decoder initialization. In typical channel coding applications,

these LLRs are computed assuming a given channel model, for example, a binary symmetric channel with known cross-over probability. For this binary channel, knowing the cross-over probability is a requirement of most soft decoding techniques, and in particular of the belief-propagation decoders of turbo codes and LDPC codes, that require it in order to compute posterior probabilities to initialize the iterative decoding process. However, this binary symmetric model turns out to be a poor match to the DSC scenario. The reason lies in the fact that the most significant bit-planes contain mostly zeros, and the probabilities that a zero in the side information becomes a one in the signal, and vice versa, are not equal. Hence, we employ an asymmetric channel model in which these probabilities are allowed to be different; denoting as  $x$  and  $y$  the channel output and input, respectively (i.e., the signal and side information), the LLRs are defined as follows:

$$\text{LLR}_X = \frac{P(x = 1 | y)}{P(x = 0 | y)} \quad \text{with } y = 0, 1, \quad (3)$$

and they are estimated considering the transmission of the bit-planes  $E^{(b)}_{Y_i}$  through a binary asymmetric channel with the following transition matrix:

$$\begin{bmatrix} 1 - p_i & p_i \\ q_i & 1 - q_i \end{bmatrix}, \quad (4)$$

where  $p_i = P(x_i = 1 | y_i = 0)$  and  $q_i = P(x_i = 0 | y_i = 1)$  are assumed to be known.<sup>1</sup> The channel output is represented by the bit-planes  $E^{(b)}_{X_i}$ . For this binary channel, the LLRs can be written as

$$\text{LLR}_X = \frac{P(x = 1 | y)}{P(x = 0 | y)} = \begin{cases} \frac{p_i}{1 - p_i} & \text{if } y = 0, \\ \frac{1 - q_i}{q_i} & \text{if } y = 1. \end{cases} \quad (5)$$

All the decoded bit-planes  $E^{(b)}_{X_i}$  are grouped together and the original prediction error array  $E_X$  is reconstructed. Finally, the decoder applies to  $E_X$  the CALIC inverse decorrelator to generate the losslessly decoded source  $X$ .

## 4. EXPERIMENTAL RESULTS: PART 1

### 4.1. Discussion

Before providing compression results on AVIRIS data, it is worth recalling and discussing a few assumptions that have been made in the implementation of DSC-CALIC.

The first assumption is that the entropy  $H_{c,i}$  of each bit-plane is available. In DSC theory,  $H(X | Y)$  is assumed to be known at the encoder of  $X$ ; this conditional entropy is a

<sup>1</sup> These probabilities can be computed by the encoder and written in the compressed file with negligible overhead. They could also be estimated by the decoder from previously decoded data, though this option has not been investigated in this paper.

measure of how much  $X$  and  $Y$  are correlated, and is needed to select the correct degree of redundancy for the S-W coder, and in practice the channel code rate of the employed LDPC code. This is a known issue in the literature, that has to be coped with. For DSC-CALIC, we assume that  $H_{c,i}$  is given for every bit-plane, and in practice it is computed from the data. To avoid this, one can, for example, let the sources communicate; this is what has been done in the second algorithm, described in Section 5. As long as the complexity of this communication is negligible, the advantages of DSC are not endangered. Otherwise, if at all possible, one can set up a feedback channel as in [2] and use punctured codes; the decoder tells the encoder to send more parity bits until decoding is correct. More recently, probability models have been proposed for estimating the entropy and other relevant probabilities with limited or no intersource communication [33]. It should be noted that this is still an open problem in the DSC literature, and that no totally satisfactory solution has been proposed so far. Related to this problem is the issue of computing the LLRs for the LDPC decoder. If the binary symmetric channel model is employed, the crossover probability must be known, while the asymmetric model requires  $p_i$  and  $q_i$ . This is another known issue, and most existing DSC techniques assume that these probabilities are known exactly; statistical estimation of these parameters could be carried out (e.g., by simply using the probabilities of the previous bit-plane, which has already been decoded), but this goes beyond the scope of this paper.

While performing channel decoding, one should check whether the decoder converged to the correct codeword, or there are some uncorrected errors. For irregular LDPC codes, if the code is slightly more powerful than necessary, as is done in DSC-CALIC, there should be no residual errors. In fact, during our experiments, the first available code whose syndrome size is larger than  $H_{c,i} \cdot N$  always decoded the signal without errors. Moreover, it is worth noticing that the convergence (or divergence) of the belief-propagation LDPC decoding process is a good (though not perfect) indicator of whether the decoding is correct, that is, the decoder has some built-in error detection capability. However, it is possible that, if not all the codes employed are equally efficient, or if some bit-planes have unusually high bit-error rate, residual errors occur after S-W decoding. The set of codes we have employed are not rate-compatible, hence it is not possible to just send more parity bits, as would be instead possible with turbo codes [2]; this is a price to pay for the higher coding efficiency of LDPC codes. We have attempted to use bit-doping, that is, sending more *systematic* bits and modifying the decoder LLRs accordingly; this strategy has turned out to be poorly matched to syndrome decoding, because too many such bits are required to make the decoding process successful. For error detection purposes, if an arbitrarily high degree of confidence is required, a CRC code can be appended to the coded bit-plane data. The CRC allows to detect decoding errors with very high probability, and the data block size is so large that the overhead due to the CRC is negligible. However, a rate-compatible code would then be required in order to avoid resending the whole bit-plane with a more powerful

code. It would be necessary to switch to a rate-compatible (but less powerful) LDPC code, or to employ a different class of codes that are rate-compatible. This issue has been left for further work.

As far as complexity is concerned, the following remarks are in order. It is known that the encoding complexity of LDPC codes is  $O(N^2)$  [34], even though the basic operations are very simple, as they are sums in modulo-2 arithmetic. The  $O(N^2)$  behavior makes the encoder complexity relatively large, and may outweigh the benefits of the DSC encoder in terms of complexity. In [34], it is noted that, however, the actual encoder complexity is quasilinear in  $N$ ; for example, for a (3,6) regular code, the number of operations can be shown to be no more than  $0.017^2 N^2 + O(N)$ . Moreover, irregular LDPC codes can be optimized so that the encoder complexity is linear. In the present work, we have decided that DSC-CALIC should not be overly concerned with complexity, but that the most efficient available codes should be used. This choice reflects the typical DSC coder designs, and allows to assess the performance loss of a practical scheme from the theoretical bounds. As will be seen in Section 4.2, comparison between the performance of DSC-CALIC and the binary and multilevel entropies allows to draw useful indications regarding the causes of performance loss, as well as the possible remedies. As a consequence, the encoding time of our software encoder is larger than that of 3D CALIC, although it could be made significantly lower by careful optimization of the set of parity check matrices.

#### 4.2. Compression of AVIRIS images

In the following, we provide compression results for DSC-CALIC, and compare it with other state-of-the-art 2D and 3D coders. The tests are performed on the 16-bit radiance data of a few AVIRIS scenes acquired in 1997; in particular, the first 256 lines (with all bands) of scene 1 of the *Cuprite*, *Lunar Lake*, and *Jasper Ridge* images have been used. The AVIRIS sensor is able to achieve a spectral resolution of about 10 nm in the visible-infrared range (400–2500 nm), thus yielding 224 highly correlated bands. We employ the publicly available radiometrically corrected radiance data, which are represented on 16 b/p, although the raw sensor data only have 12 b/p. Since both scalar and vector codecs consider one pair of bands at a time. For all the algorithms considered in this paper, the data are assumed to be available in band-sequential (BSQ) format.

The following algorithms have been compared. JPEG-LS [35] has been considered, as it is an efficient and low-complexity technique. 2D-CALIC [29] and 3D-CALIC [36] are more complex, but provide state-of-the-art performance. Specifically, 2D-CALIC and JPEG-LS are two examples of codecs exploiting only intraband correlation, whereas 3D-CALIC performs both spatial and spectral decorrelation of the hyperspectral cube.

The technique labeled as “B-iDSC-CALIC” is a version of DSC-CALIC in which S-W coding is ideal, that is, assuming that the S-W coder employs a bit rate exactly equal to the conditional entropy; the initial letter “B” indicates that

TABLE 1: Coding efficiency of DSC-CALIC compared to that of conventional 2D and 3D schemes.

Algorithm	<i>Cuprite</i>	<i>Lunar</i>	<i>Jasper</i>	<i>Average</i>
JPEG-LS	6.78	6.98	7.58	7.11
2D-CALIC	6.61	6.84	7.43	6.96
3D-CALIC	5.11	5.25	5.13	5.16
DSC-CALIC	5.99	6.22	6.37	6.19
B-iDSC-CALIC	5.94	6.17	6.31	6.14
M-iDSC-CALIC	5.12	5.21	5.13	5.15

S-W coding is binary. A practical S-W coder, such as DSC-CALIC, cannot achieve a rate smaller than the conditional entropy, that is, smaller than that of B-iDSC-CALIC, in the same way as a Huffman or arithmetic coder can not achieve a rate smaller than the source entropy. As a consequence, B-iDSC-CALIC can be used to evaluate the performance loss of the practical LDPC-based S-W encoder with respect to the ideal case.

Similarly, the technique labeled as “M-iDSC-CALIC” assumes ideal S-W coding; however, in this case, the entropy is not computed considering bit-planes as binary sources and summing their entropies, hence disregarding the correlation among different bit-planes. Rather, we consider the complete multilevel source, with an alphabet containing  $2^{16}$  symbols, and compute the frequency of occurrence of each symbol. The multilevel entropy represents the performance bound for a multilevel as opposed to a binary S-W coder.

The results are reported in Table 1. As can be seen, DSC-CALIC obtains significantly lower bit rates than all 2D coding techniques, and in particular is on average 0.77 b/p better than 2D-CALIC and 0.92 b/p better than JPEG-LS, notwithstanding that DSC-CALIC does not perform context modeling, which amounts to a performance loss of about 0.2 b/p.

Comparing DSC-CALIC and iDSC-CALIC, it can also be seen that the proposed S-W coder is very efficient, in that it is only about 0.05 bit worse than an ideal bit-plane-based S-W coder.

Comparing iDSC-CALIC and 3D CALIC, it can be seen that even the ideal S-W coder falls short of the theoretical bounds, in that there is still a gap with respect to a 3D technique. Part of this gap is due to the context modeling in 3D-CALIC, which adaptively captures spatial and spectral features, and does not neglect correlation between different bit-planes. This can be further explained by looking at the entropies for multilevel S-W coding. Although the average multilevel entropy for each band is only slightly less than the binary entropy (0.08 b/p), the multilevel conditional entropy is 1 b/p lower than the binary one. This is the reason why 3D-CALIC can get extremely close to the multilevel entropy, while DSC-CALIC and iDSC-CALIC are 1 b/p away. The difference between the binary and multilevel conditional entropies highlights a limit of binary S-W coders, which are not able to capture all the correlation in the data. Although DSC-CALIC is extremely close to iDSC-CALIC, significantly better performance could be obtained by using a multilevel

coder, or at least a binary coder whose posterior probabilities are computed taking into account the already decoded bit-planes. In fact, as will be seen in Section 6, a scalar multilevel coder may achieve performance as good as a capacity-achieving binary code, with much lower complexity.

## 5. SPATIALLY ADAPTIVE DSC OF HYPERSPECTRAL IMAGES

To develop the DSC coder described in the previous section, we assumed that the quantity  $H(X | Y)$  is known. A possible solution could be to estimate  $H(X | Y)$  “on the fly,” however this may require too much communication between image bands. The problem is further complicated by the observation that image bands can not be modelled as stationary sources hence calling for an adaptive estimation of  $H(X | Y)$ .

In this section, we present a second DSC codec that tackles the above problem by relying on a completely different strategy than DSC-CALIC. First of all, we allow the encoder to compare two consecutive image bands, say  $X$  and  $Y$ . However, such a comparison is kept as simple as possible, thus permitting to achieve a considerable gain in terms of encoder simplicity, at the expense of a symmetric complication of the decoder. Secondly, we adopt a block-based approach that, by splitting the image bands into small parts, allows to finely adapt the estimation of image dependency. This forces us to use codes having a small length, thus tending to reduce the effectiveness of the system. As it will be shown in the next section, though, the benefits brought by adaptivity outperform the coding loss, thus permitting to achieve rather good coding efficiency.

In order to describe the DSC codec, let us focus on two consecutive bands  $X$  and  $Y$ . We assume that  $Y$  has already been coded losslessly and hence it is available at the decoder. To see how  $X$  can be coded, let us assume that a model  $\mathcal{P}$  permitting to predict the value of the pixels in  $X$  from those of  $Y$  is known (we will remove such an assumption later on), that is, the value of each pixel  $x(i, j)$  in  $X$  can be written as

$$x(i, j) = \mathcal{P}(y(i, j)) + n(i, j), \quad (6)$$

where  $n(i, j)$  is called correlation noise, and  $y(i, j)$  is the value of the pixel in  $Y$ . Of course, using a scalar predictor as in (6) is not an optimum choice, however we opted for it in order to maximize the simplicity of the encoder. We also assume that the statistics of  $n(i, j)$  are known. The value of the pixels in  $X$  is coded by retaining only the  $k$  least significant bits of  $x(i, j)$ , where the exact value of  $k$  depends on the correlation between the bands, that is, on the statistics of  $n$ . Note that these  $k$  least significant bits (LSB) can be seen as the index  $i^*$  of the coset containing  $x(i, j)$ . Specifically, each coset is formed by all the grey levels with the same LSBs, and each codeword in the coset is characterized by the particular values assumed by the most significant bits (MSB). The decoder will have to resolve this ambiguity by recovering the MSBs on the basis of the side information  $Y$ . Specifically, it chooses the most significant bits to minimize the distance between the reconstructed pixel value and the predicted value

Number $k$ of transmitted bits	$K$ least significant bit-planes	CRC bits
-----------------------------------	-------------------------------------	----------

FIGURE 2: Content of a compressed block for the adaptive DSC scheme.

$\mathcal{P}(y(i, j))$ . If the number of transmitted bits is chosen properly, no reconstruction loss is incurred.

The main problem with the above implementation is that the model  $\mathcal{P}$  and the statistics of  $n(i, j)$  are not known, and, in any case, are not stationary quantities. To get around the problem, the availability of  $Y$  at the encoder is exploited. Note that being our final aim that of reducing the complexity of the encoder, the availability of  $Y$  can still be exploited, as long as we do not increase the encoder complexity. Specifically, we first split  $X$  into nonoverlapping blocks of size  $n \times m$ . Then, for each block, a rough estimate of the maximum value of  $n(i, j)$  is computed and used to decide the number of bits that can be discarded while coding the pixels of the block.

In addition to the least significant bits of  $X$ , the encoder computes some parity check bits by applying a CRC code to the values of the pixels in the block (the reason for the introduction of the CRC bits will be explained below). Finally, the encoder specifies the number of least significant bits actually stored. The exact content of the coded bit stream is summarized in Figure 2.

Since the decoder does not know the particular  $\mathcal{P}$  used by the encoder, it considers several predictors obtaining, for each pixel, a set of  $n_p$  possible predicted values

$$\tilde{x}_l(i, j) = \mathcal{P}_l(y(i, j)), \quad l = 1, \dots, n_p. \quad (7)$$

When the MSBs of the predicted values are combined with the least significant bits of  $x(i, j)$ ,  $n_p$  possible reconstructed values are obtained. The decoder uses the CRC bits computed by the encoder to choose the right predictor. In particular, it looks for a  $\mathcal{P}$  for which the reconstructed values of the pixels in the block result in the correct CRC sequence. If the length of the CRC sequence is chosen properly, the probability that the correct parity bits are obtained for a wrong reconstruction can be made arbitrarily small, hence ensuring the lossless reconstruction of  $X$ .

### 5.1. Estimation of $k$

A major problem at the encoder side is that of defining the number  $k$  of LSBs to be transmitted at the decoder. To describe the strategy we developed, let us start by observing that, since two adjacent bands of a hyperspectral image generally exhibit a strong correlation, the correlation coefficient  $\rho_{XY}$  between a block of the band  $X$  and the corresponding block in the previous band  $Y$  is very close to 1. As a consequence, there exists an approximately linear dependency between the two blocks and a linear prediction model based on the side information  $Y$  is likely to be a good predictor for the current block to be coded. To be specific, let us focus on a single block in  $X$ . Let  $\mu_x$  be the average value of the block. The encoder considers a set of linear predictors leading to a

pool of predicted values  $\{\tilde{x}^{(l)}(i, j)\}$

$$\tilde{x}^{(l)}(i, j) = \mu_x + \alpha^{(l)}[y(i, j) - \mu_y], \quad (8)$$

where  $\mu_y$  is the average value of the corresponding block in  $Y$  and  $\alpha^{(l)}$  is a varying parameter. In particular,  $\alpha^{(l)}$  is determined by inserting within (8) the true values of a pair of corresponding pixels in  $X$  and  $Y$ . So, given two pixels  $x(i_l, j_l)$  and  $y(i_l, j_l)$ , the parameter  $\alpha^{(l)}$  is determined as

$$\alpha^{(l)} = \frac{x(i_l, j_l) - \mu_x}{y(i_l, j_l) - \mu_y}. \quad (9)$$

In our implementation, the pixels  $x(i_l, j_l)$  and  $y(i_l, j_l)$  are all the pixels on the left and upper borders of the to-be-coded block, that is, all the pixels belonging either to the first row or to the first column of the block (only one pixel out of four is considered so to reduce as much as possible the computational complexity of the encoder). Eventually, by indicating with  $\mathbf{x}$  the vector with all the values of the pixels in the block, the quantity

$$n_{\min} = \min_l \|\mathbf{x} - \tilde{\mathbf{x}}^{(l)}\|_{\infty} \quad (10)$$

is taken as a rough estimate of the correlation noise existing between the current block  $\mathbf{x}$  and the corresponding block in  $Y$ . This quantity is used to set the number of least significant bits to be retained as

$$k = \lceil \log_2 n_{\min} \rceil + 1, \quad (11)$$

that is, the minimum number of least significant bits to be transmitted for each pixel that ensures a lossless recovery at the decoder.

### 5.2. Decoding

In order to exploit the side information to recover the most significant bits of  $X$ , the decoder would need to know a model to predict the values of the pixels in  $X$  from those in  $Y$ . This model does not need to be the same used by the encoder, it is only required that the prediction error experienced by the decoder does not exceed the maximum prediction error estimated by the encoder. For sake of simplicity, we decided to use a predictor having the same form of the one used by the encoder. However, the decoder does not know the average value  $\mu_x$  of the current block, nor the value of the pixels in the first row and in the first column. Then it has to estimate these parameter, by resorting to the pixels of spatially contiguous blocks already decoded and to those belonging to the side information  $Y$ .

Let us call  $\mu_{xu}$  and  $\mu_{xl}$  the average values of the blocks above and on the left of the current block, and  $\mu_{yu}$  and  $\mu_{yl}$  the corresponding values in the previous band. By assuming that the spatial correlation in  $Y$  is retained in  $X$ , we use the



following estimate for the mean value  $\mu_x$ :

$$\hat{\mu}_x = \frac{1}{2} \left[ \mu_{xu} \left( 1 + \frac{\mu_y - \mu_{yu}}{\mu_{yu}} \right) + \mu_{xl} \left( 1 + \frac{\mu_y - \mu_{yl}}{\mu_{yl}} \right) \right], \quad (12)$$

which is more and more accurate as  $\rho_{XY}$  gets closer to 1. A similar expression is used to estimate the values of all the pixels of the first row and the first column of the block to be decoded.

The estimates of the pixels in the border of the block to be decoded undergo a further refinement obtained by replacing their least significant bits with those received from the encoder. The estimate of  $\mu_x$  and those of the border pixels are used to define a set of predictors  $\mathcal{P}_l$ . Indeed, a larger number of predictors is built by perturbing the estimated values of  $\mu_x$  and those of the border pixels by adding a number of multiples of a given quantization step. Of course, the lower the quantization step is the more precise, and the more complex, the decoder is.

All the candidate predicted values obtained by applying  $\mathcal{P}_l$  to  $Y$  are combined with the received least significant bits resulting in a right reconstruction of the current block if the condition

$$|\hat{x}^{(l)}(i, j) - x(i, j)| < 2^{k-1} \quad (13)$$

is verified for all the pixels in the block. In fact, a coset index of length  $k$  specifies a coset of values at a distance of  $2^k$  from each other and in order to select the correct value  $x(i, j)$  the predicted value  $\hat{x}^{(l)}(i, j)$  must be at a distance lower than  $2^{k-1}$ .

### 5.3. Vector extension

#### 5.3.1. Vector encoder

We also implemented a vector extension of the above scheme. In fact, using a vector encoder permits to increase the minimum distance between two elements of the coset, thus providing a higher immunity against spectral correlation noise than in the scalar case. More in detail, the coset index is formed by two parts, the former obtained scalarly by retaining the  $k-1$  least significant bit-planes of the block, as in the scalar coder, the latter, let us indicate it by  $s_k$ , computed by applying channel coding to the  $k$ th bit-plane (see Figure 3). In order to determine the number of bit planes to be transmitted, the encoder makes an estimate of the correlation noise existing between the current block and the corresponding block in the previous band, by using the same approach adopted for the scalar encoder. Note that the length of  $s_k$  is at the most equal to the length of the bit-plane itself. Since the dimension of a block is fixed to  $2^q - 1$  pixels (for some  $q > 0$ ), the bit-plane has  $2^q - 1$  bits thus allowing to employ a  $(2^q - 1, 2^q - 1 - l_s)$  BCH channel code [37] to transmit a syndrome of length  $l_s$ :

$$s_k = p_k \mathcal{H}^t, \quad (14)$$

where  $p_k$  is the  $k$ th bit-plane and  $\mathcal{H}$  is the  $l_s \times (2^q - 1)$  parity check matrix of the BCH code. The length  $l_s$  of the syndrome

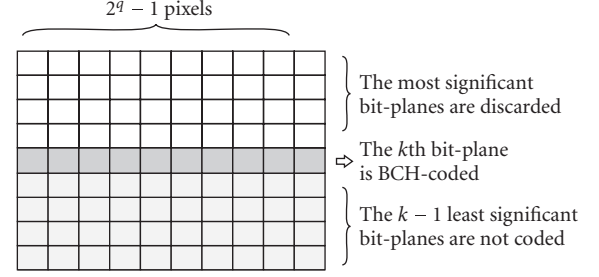


FIGURE 3: Bit-plane coding by means of the vector DSC coder. The  $k$ th bit-plane is BCH-coded, whereas the  $k-1$  least significant bit-planes are transmitted uncoded.

$s_k$  is preliminarily estimated in order to select the BCH code with an error correction capability matched to the correlation noise and to ensure a lossless reconstruction at the decoder. In summary, the bit-stream relative to a compressed block includes

- (1)  $(k-1) \times r \times c$  bits for the  $(k-1)$  least significant bit planes ( $r$  and  $c$  are the dimensions of the block);
- (2)  $\lceil \log_2(h-1) \rceil$  bits to code the scalar syndrome length (having indicated with  $h$  the number of bits each pixel value consists of);
- (3)  $l_s$  bits of the vectorial syndrome  $s_k$  relative to the  $k$ th least significant bit-plane of the block;
- (4)  $(q-2)$  bits to code the vectorial syndrome length, as the maximum error correction capability of a BCH code of length  $(2^q - 1)$  is typically  $(2^{q-2} - 1)$  [37];
- (5)  $p$  parity check bits relative to the CRC computed on the block.

The choice of using a channel code to compress  $X$  is consistent with DSC theory which sees the side information  $Y$  (or, as in our case, the prediction of  $X$  obtained by relying on  $Y$ ) as the output of a virtual channel with  $X$  as an input. To correct the errors induced by this correlation noise, the decoder needs some redundant information which is nothing but the  $X$ -syndrome delivered by the BCH coder.

#### 5.3.2. Vector decoder

The decoder resorts to a suitable linear prediction estimate obtained by means of the side information and the already decoded bits as done for the scalar case. For each candidate predictor, the decoder selects among the elements of the coset specified by the syndrome  $s_k$  the bit-plane which is at minimum Hamming distance from the  $k$ th bit-plane of the predictor currently tested. A successful decoding is obtained if the number of errors induced by a candidate predictor is lower than the error correction capability of the BCH code. Regardless of the result of the decoding of the  $k$ th bit-plane, the  $k-1$  least significant bit-planes are added to form  $2^q - 1$  scalar syndromes of length  $k$ . Then the decoder needs to determine the most significant bits of each pixel. As in the scalar case, this is done by minimizing the distance to the predicted pixel value. The reconstruction of the current block is

TABLE 2: Coding efficiency of the adaptive DSC coders compared to that of conventional 2D and 3D schemes.

Algorithm	<i>Cuprite</i>	<i>Lunar</i>	<i>Jasper</i>	Average
JPEG-LS	6.78	6.98	7.58	7.11
2D-CALIC	6.61	6.84	7.43	6.96
3D-CALIC	5.11	5.25	5.13	5.16
s-DSC	6.08	6.23	6.24	6.18
v-DSC	5.94	6.09	6.10	6.04

considered to be successful if the CRC sequence computed on the reconstructed pixels matches the received CRC, otherwise another candidate predictor is tested.

## 6. EXPERIMENTAL RESULTS: PART 2

To check the effectiveness of the adaptive schemes both from the point of view of computational complexity and coding efficiency, the scalar and the vector DSC adaptive codecs have been applied to the same hyperspectral images used in Section 4.2, and the results we obtained have been compared to those of existing popular lossless compression algorithms, namely, 2D-CALIC, JPEG-LS, and 3D-CALIC. As in the previous case, all data are assumed to be available in BSQ format.

With regard to the scalar codec, we chose a block size of  $16 \times 16$ , since it provides a good balance between adaptivity and length of the channel code. Note also that the use of a small block size increases the relative weight of headers and trailers carrying the information about syndrome length and checksum. As to the vector codec, a  $15 \times 17$  block is employed allowing to use a  $(255, 255 - l_s)$  BCH code ( $q = 8$ ), while a CRC code with the same parity check length ( $p = 32$ ) is adopted for both the DSC codecs, thus providing a decoding error probability of  $2^{-32}$ .

All the algorithms have been run on a workstation with a Pentium III 850 MHz processor and Linux 2.6.5 operating system. Table 2 reports the bit rates achieved by each compression algorithm for *Cuprite*, *Lunar*, and *Jasper* images as well as the average value relating to the three scenes, while Figure 4 depicts the computing times per band, averaged over the three images for both encoder and decoder. As expected, both DSC schemes show a remarkable asymmetry in the computational complexity between encoder and decoder, the former resulting much faster than the latter. By comparing the new schemes, there is a little gain in the compression ratio (nearly 0.1 b/p) achieved by the vector scheme with respect to the scalar one though it causes a considerable increase of the computational cost both in coding and decoding operation. A comparison with the other methods reveals that the DSC codecs perform about 1 b/p below the bit rates achieved by JPEG-LS and 2D-CALIC but about 1 b/p above the bit rates of 3D-CALIC. On the other side, the scalar DSC encoder, which is the fastest between the two DSC schemes, is about 10 times faster than 3D-CALIC and about 5 times faster than 2D-CALIC, though it is about 2 times slower than JPEG-LS.

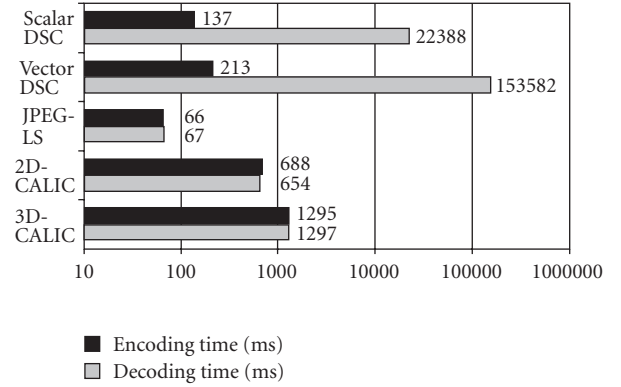


FIGURE 4: Computational complexities of DSC and conventional codecs. Both the per-band complexities of the encoder and the decoder are reported.

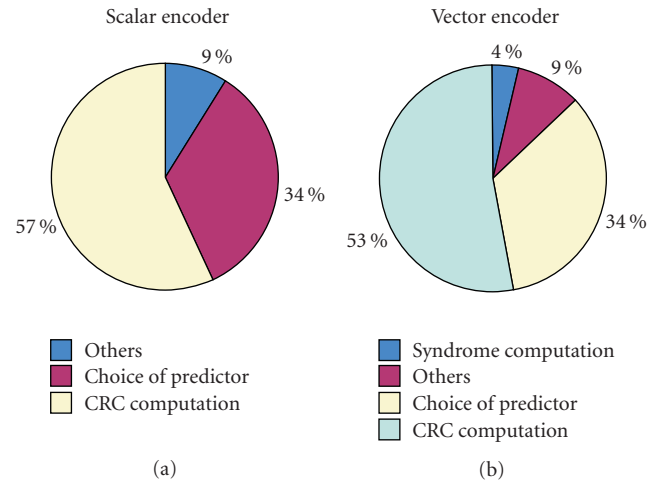


FIGURE 5: Computational complexity of the scalar and vector encoders split among tasks.

Comparing the results in Table 2 with those of DSC-CALIC in Table 1, it can be seen that the average compression performance of DSC-CALIC is similar to that of s-DSC. The vector extension has improved performance, but also improved complexity. Therefore, it turns out that, for the application considered in this paper, a scalar multilevel code achieves the same performance as a powerful binary code, with evident advantages in terms of encoder computational complexity.

### 6.1. Further insights on the encoder complexity

In order to better understand where the residual complexity of the encoder derives from, we measured the impact that the various steps performed by the encoder have on the encoding time. The results we obtained are shown in Figure 5.

Interestingly, the calculation of the CRC is the most computationally intensive operation. At the same time, the choice of the predictor, thanks to the particular procedure

we adopted that estimates it by considering only a small subset of the pixels in the block, accounts only of one third of the computing time. The above observation suggests some possible directions to further speed up the encoder. First of all, we can observe that the error correction code we used (a 255 long BCH code) offers good geometric properties but is characterized by a rather complicated encoding process. Faster computation of CRC bits could be achieved by adopting a simpler code, possibly at the expense of worse error detection properties.<sup>2</sup> A second way of improving the system consists in deciding to transmit the parameters of the predictor used by the encoder. In this way, we could avoid to transmit the CRC bits hence significantly speeding up the encoder. This improvement would be paid by a slight increase of the bit rate, since storing the predictor parameters requires a few more bits than the CRC length. It is also worth noting that, at least in the scalar case, this latter approach would constitute a significant deviation from the DSC paradigm, since the new scheme would closely resemble a classical interband prediction-based encoder.

## 7. DISCUSSION AND CONCLUSIONS

The objective of this paper was to evaluate the potential of DSC as applied to real-world data in a realistic scenario. Among the possible benefits brought by the DSC approach, we focused on reducing the computational burden of the encoder. Specifically, our target application was on-board lossless compression of hyperspectral data with reduced encoder complexity. The possibility of exploiting the opportunities offered by DSC theory to improve other aspects of hyperspectral codecs, for example, to reduce the communication bandwidth between the two sources of information represented here by two consecutive image bands, is left for future research.

To achieve the above objective, we have designed two DSC coders that borrow from existing coders as for the pure S-W coding, but exhibit distinctive features in their application to real-world data. The rationale behind designing two coders is to evaluate two different approaches to DSC, that is, capacity achieving channel codes, and simple multilevel codes. The performance of these techniques has been compared with that of state-of-the-art 2D and 3D lossless compression algorithms. The following remarks can be made on the basis of the experimental results on AVIRIS data.

- (i) Practical binary S-W coding based on LDPC codes has very high performance, since it is as close as 0.05 b/p to the theoretical bound.
- (ii) However, the encoder complexity of capacity-achieving binary codes is not negligible, and may outweigh the benefits of DSC if the encoder is not implemented carefully.
- (iii) Binary S-W coding turns out to be significantly sub-optimal with respect to multilevel S-W coding. This

can be seen by comparing binary and multilevel conditional entropies, which differ by about 1 b/p.

- (iv) This implies that a simple multilevel coset code can obtain performance as good as a powerful binary code, with much lower complexity. This result is achieved by the proposed scalar scheme. In fact, the performance of the scalar scheme is equivalent to that of DSC-CALIC, but with much lower complexity.
- (v) This also implies that there is still a lot of performance gain to be achieved, in order for DSC-based schemes to get as close as possible to the performance of a multilevel joint encoder such as 3D-CALIC. This can be done by employing capacity-achieving multilevel channel codes. However, in [38] it is shown that the most significant correlation is between the current and the previous bit-plane; this suggests that careful optimization of a binary coder, employing a probability model that takes into account also the probability of the symbols in the previous bit-plane, might lead to improved performance. Whether this also holds for 16-bit data is still to be investigated.

We would like conclude our discussion by explicitly observing that the schemes we developed, though offering promising results, are not efficient enough to compete with classical state-of-the-art techniques. Indeed, similar or better performance, both in terms of coding efficiency and simplicity, can be achieved by applying a very simple interband prediction scheme followed by intraband coding of the prediction error (e.g., by means of a standard JPEG-LS encoder). This is not surprising: it took several years before conventional lossless interband prediction-based codecs reached the maturity level they offer today. Similarly, it is easy to imagine that a significant amount of research is still needed before encoders designed according to the DSC paradigm can compete with their predecessors. Nevertheless, we deem that our investigation is an interesting one, because it clearly highlights the difficulties encountered when trying to apply DSC techniques to remote sensing image compression and, most of all, it introduces a bunch of new ideas that permit to reduce the gap between theoretically predicted performance and practical schemes.

## ACKNOWLEDGMENT

The research presented in this paper was partially funded by the University of Siena under Project PAR-2005, entitled “Analysis and development of distributed source coding techniques for remote sensing applications.”

## REFERENCES

- [1] Z. Xiong, A. D. Liveris, and S. Cheng, “Distributed source coding for sensor networks,” *IEEE Signal Processing Magazine*, vol. 21, no. 5, pp. 80–94, 2004.
- [2] B. Girod, A. M. Aaron, S. Rane, and D. Rebollo-Monedero, “Distributed video coding,” *Proceedings of the IEEE*, vol. 93, no. 1, pp. 71–83, 2005.
- [3] D. Slepian and J. K. Wolf, “Noiseless coding of correlated information sources,” *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, 1973.

<sup>2</sup> Worse error detection properties, in turn, call for a longer code, hence slightly diminishing the compression efficiency.

- [4] A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 1–10, 1976.
- [5] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, New York, NY, USA, 1991.
- [6] A. Nonnis, M. Grangetto, E. Magli, G. Olmo, and M. Barni, "Improved low-complexity intraband lossless compression of hyperspectral images by means of Slepian-Wolf coding," in *Proceedings of IEEE International Conference on Image Processing (ICIP '05)*, vol. 1, pp. 829–832, Genova, Italy, September 2005.
- [7] M. Barni, D. Papini, A. Abrardo, and E. Magli, "Distributed source coding of hyperspectral images," in *Proceedings of IEEE International Geoscience and Remote Sensing Symposium (IGARSS '05)*, vol. 1, pp. 120–123, Seoul, Korea, July 2005.
- [8] C. Tang, N.-M. Cheung, A. Ortega, and C. S. Raghavendra, "Efficient inter-band prediction and wavelet based compression for hyperspectral imagery: a distributed source coding approach," in *Proceedings of Data Compression Conference (DCC '05)*, pp. 437–446, Snowbird, Utah, USA, March 2005.
- [9] X. Li, "Distributed coding of multispectral images: a set theoretic approach," in *Proceedings of IEEE International Conference on Image Processing (ICIP '04)*, vol. 2, pp. 3105–3108, Singapore, October 2004.
- [10] R. Puri and K. Ramchandran, "PRISM: a "reversed" multimedia coding paradigm," in *Proceedings of IEEE International Conference on Image Processing (ICIP '03)*, vol. 1, pp. 617–620, Barcelona, Spain, September 2003.
- [11] M. Gastpar, P. L. Dragotti, and M. Vetterli, "On compression using the distributed Karhunen-Loève transform," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04)*, vol. 3, pp. 901–904, Montreal, Quebec, Canada, May 2004.
- [12] S. S. Pradhan, J. Chou, and K. Ramchandran, "Duality between source coding and channel coding and its extension to the side information case," *IEEE Transactions on Information Theory*, vol. 49, no. 5, pp. 1181–1203, 2003.
- [13] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): design and construction," *IEEE Transactions on Information Theory*, vol. 49, no. 3, pp. 626–643, 2003.
- [14] J. Garcia-Frias and Y. Zhao, "Compression of correlated binary sources using turbo codes," *IEEE Communications Letters*, vol. 5, no. 10, pp. 417–419, 2001.
- [15] A. D. Liveris, Z. Xiong, and C. N. Georgiades, "Distributed compression of binary sources using conventional parallel and serial concatenated convolutional codes," in *Proceedings of Data Compression Conference (DCC '03)*, pp. 193–202, Snowbird, Utah, USA, March 2003.
- [16] Y. Yang, S. Cheng, Z. Xiong, and W. Zhao, "Wyner-Ziv coding based on TCQ and LDPC codes," in *Proceedings of the 37th Asilomar Conference on Signals, Systems, and Computers*, vol. 1, pp. 825–829, Pacific Grove, Calif, USA, November 2003.
- [17] A. Majumdar, J. Chou, and K. Ramchandran, "Robust distributed video compression based on multilevel coset codes," in *Proceedings of the 37th Asilomar Conference on Signals, Systems, and Computers*, vol. 1, pp. 845–849, Pacific Grove, Calif, USA, November 2003.
- [18] A. D. Liveris, Z. Xiong, and C. N. Georgiades, "A distributed source coding technique for correlated images using turbo-codes," *IEEE Communications Letters*, vol. 6, no. 9, pp. 379–381, 2002.
- [19] Q. Xu and Z. Xiong, "Layered Wyner-Ziv video coding," in *Visual Communications and Image Processing*, vol. 5308 of *Proceedings of SPIE*, pp. 83–91, San Jose, Calif, USA, January 2004.
- [20] H. Wang and A. Ortega, "Scalable predictive coding by nested quantization with layered side information," in *Proceedings of IEEE International Conference on Image Processing (ICIP '04)*, vol. 3, pp. 1755–1758, Singapore, October 2004.
- [21] A. Sehgal, A. Jagmohan, and N. Ahuja, "Wyner-Ziv coding of video: an error-resilient compression framework," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 249–258, 2004.
- [22] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, 1996.
- [23] M. Grangetto, E. Magli, and G. Olmo, "Context-based distributed wavelet video coding," in *Proceedings of IEEE International Workshop on Multimedia Signal Processing (MMSP '05)*, Shanghai, China, October–November 2005.
- [24] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*, Kluwer Academic, Boston, Mass, USA, 2001.
- [25] B. Penna, T. Tillo, E. Magli, and G. Olmo, "Progressive 3-D coding of hyperspectral images based on JPEG 2000," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 1, pp. 125–129, 2006.
- [26] R. Gallager, *Low Density Parity Check Codes*, MIT Press, Cambridge, Mass, USA, 1963.
- [27] A. D. Liveris, Z. Xiong, and C. N. Georgiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Communications Letters*, vol. 6, no. 10, pp. 440–442, 2002.
- [28] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1261–1271, 1996.
- [29] X. Wu and N. Memon, "Context-based, adaptive, lossless image coding," *IEEE Transactions on Communications*, vol. 45, no. 4, pp. 437–444, 1997.
- [30] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 399–431, 1999.
- [31] S.-Y. Chung, G. D. Forney Jr., T. J. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Communications Letters*, vol. 5, no. 2, pp. 58–60, 2001.
- [32] A. Moffat, R. M. Neal, and I. H. Witten, "Arithmetic coding revisited," *ACM Transactions on Information Systems*, vol. 16, no. 3, pp. 256–294, 1998.
- [33] N.-M. Cheung, H. Wang, and A. Ortega, "Correlation estimation for distributed source coding under information exchange constraints," in *Proceedings of IEEE International Conference on Image Processing (ICIP '05)*, vol. 2, pp. 682–685, Genova, Italy, September 2005.
- [34] T. J. Richardson and R. L. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 638–656, 2001.
- [35] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS," *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1309–1324, 2000.



- [36] X. Wu and N. Memon, "Context-based lossless interband compression-extending CALIC," *IEEE Transactions on Image Processing*, vol. 9, no. 6, pp. 994–1001, 2000.
- [37] J. G. Proakis and M. Salehi, *Communication Systems Engineering*, Prentice-Hall, Upper Saddle River, NJ, USA, 2nd edition, 2002.
- [38] S. Cheng and Z. Xiong, "Successive refinement for the Wyner-Ziv problem and layered code design," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, part 2, pp. 3269–3281, 2005.

**Enrico Magli** received the M.S. degree in electronics engineering in 1997, and the Ph.D. degree in electrical engineering in 2001, from Politecnico di Torino, Turin, Italy. He is currently an Assistant Professor at the same university. His research interests are in the fields of error-resilient image and video coding for wireless applications, compression of remote sensing images, image security and digital watermarking, and distributed source coding. From March to August 2000, he was a Visiting Researcher at the Signal Processing Laboratory of the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland. He has coauthored more than 100 scientific papers in international journals and conferences. He is currently a Member of the Data Archiving and Distribution (DAD) Technical Committee of the IEEE Geoscience and Remote Sensing Society, of the Multimedia Systems and Applications (MSA) Technical Committee of the IEEE Circuits and Systems Society, and coeditor of JPEG 2000 Part 11 (*Wireless Applications*). He has been a Member of the Technical Program Committee, Session Chair and Special Session Organizer for several international conferences, including IEEE ICASSP, ICME, ICIP, MMSP, ISCAS, and IGARSS, and is in the editorial board of the EURASIP Journal on Information Security.



**Mauro Barni** graduated in electronic engineering at the University of Florence in 1991. He received the Ph.D. degree in informatics and telecommunications in October 1995. He currently carries out his research activity at the Department of Information Engineering of the University of Siena where he works as an Associate Professor. His main research focus regards the application of image processing techniques to copyright protection and authentication (digital watermarking). He is also interested in DSC theory. He is the author/coauthor of about 170 papers published in international journals and conference proceedings, and holds three patents in the field of digital watermarking. He is the coauthor of the book *"Watermarking Systems Engineering: Enabling Digital Assets Security and other Applications"*, published by Dekker Inc. He was the Chairman of the IEEE Multimedia Signal Processing Workshop held in Siena in 2004, and the chairman of the IV edition of the International Workshop on Digital Watermarking. He is the editor in chief of the EURASIP Journal on Information Security and serves as an Associate Editor of several international journals. He is a Member of the IEEE Information Forensic and Security technical Committee (IFS-TC) of the IEEE Signal Processing Society. He is a Senior Member of the IEEE and EURASIP.



**Andrea Abrardo** graduated cum laude in 1993 in electronic engineering at the University of Florence, Italy, discussing a thesis on "differential demodulation and decoding for radiomobile channels." From January 1994 to November 1994, he has worked in the Image Processing and Communications Laboratory of the Department of Electronic Engineering of the University of Florence, collaborating with the Tuscany Region for the development of broadband networks infrastructures. From November 1994 to October 1997, he attended the Ph.D. degree at the same Department of Electronic Engineering. In June 1998, he got the Ph.D. degree discussing a thesis on "web-based teleradiology systems." In 1998, he joined the Department of Informatic Engineering of the University of Siena, Italy, as a Researcher. His current position at the Department of Informatic Engineering of the University of Siena is Associate Professor.



**Marco Grangetto** received the "summa cum laude" degree in electrical engineering in 1999, and the Ph.D. degree in 2003, from Politecnico di Torino, Italy. He is currently a postdoctoral Researcher at the Image Processing Lab, in the same University. He was awarded the Premio Optime by Unione Industriale di Torino in September 2000, and a Fulbright Grant in 2001 for a research period with the Department of Electrical and Computer Engineering, University of California, San Diego. His research interests are in the fields of multimedia signal processing and communications. In particular, his expertise includes wavelets, image and video codings, data compression, video error concealment, error-resilient video coding unequal error protection, and joint source channel coding. He is participating to the ISO standardization activities on JPEG 2000, where he is a contributor to Part 11 of the standard. He has been a Member of the Technical Program Committee for several international conferences, including IEEE ICME, ICIP, and ISCAS.

